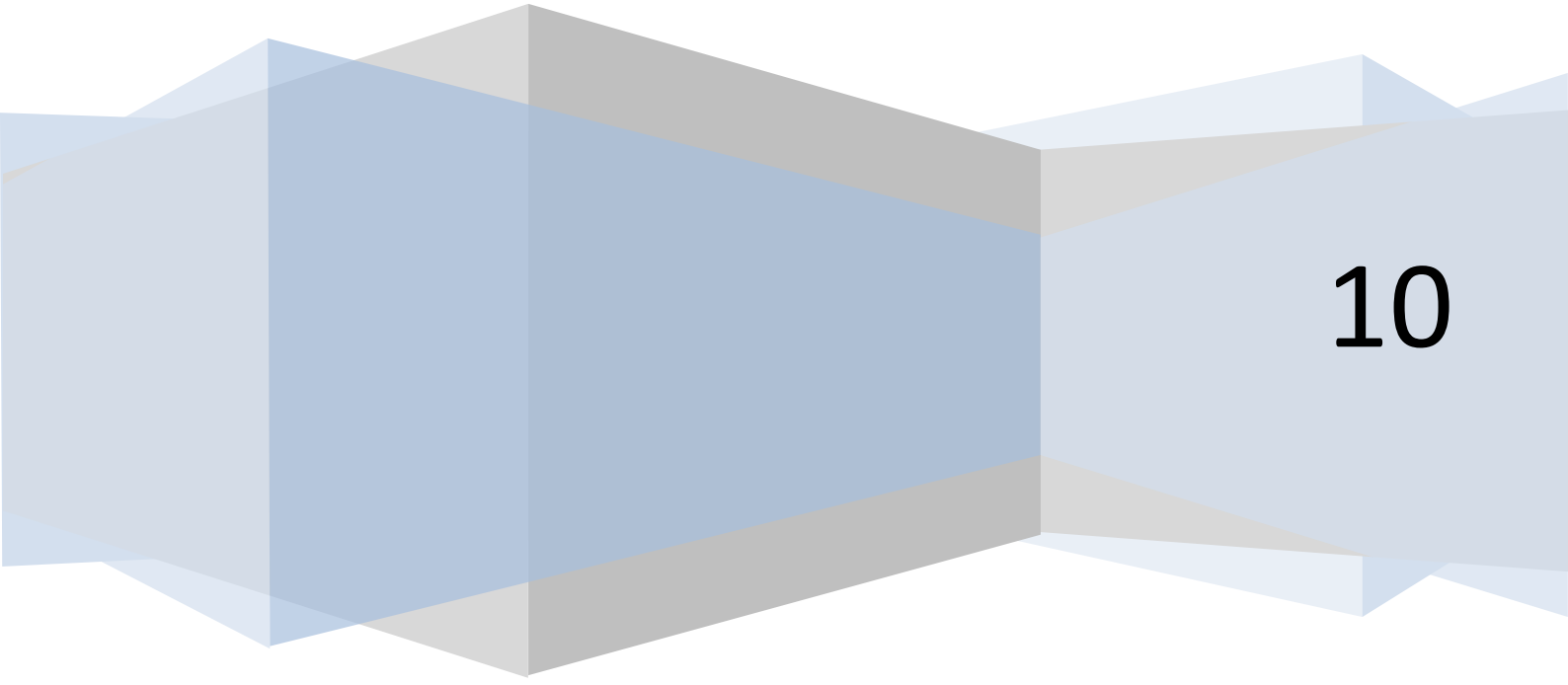
EISTI

Projet Java EE Approfondi

Manuel d'installation du framework Stripes

Amaury Languillat, Yann Gonzalez, Arnaud Recher, Vincent Laronde, Anys Mechkar



10

Manuel d'installation

Téléchargement

On part du principe que l'utilisateur possède :

- La version Ganymede 3.4.2 (au minimum) d'Eclipse pour les développeurs Java EE
- La version 6.0.18 (au minimum) du conteneur JEE Apache Tomcat

Voici les fichiers nécessaires pour l'installation de Stripes : Stripes.zip. On y trouve :

- Toute la documentation nécéssaire.
- Le fichier Properties
- Les bibliothèques

Création d'un projet JEE sous éclipse

Nous partons du principe que l'utilisateur est en mesure de créer un projet de type : Dynamic Web Project. Dans les options, il faut choisir la dernière version du module Dynamic Web : 2.5 et la bonne configuration du serveur : Apache Tomcat v6.0.

Votre projet JEE est maintenant créé et constitue la base de votre futur projet basé sur le framework Stripes 1.5.1.

Ajout de Stripes au projet

Dans un premier temps, vous devez copiez les 3 .jar du dossier "/lib/" du zip de Stripes que vous avez téléchargé dans "/WEB-INF/lib/". Voici leurs noms :

- stripes.jar
- commons-logging.jar
- cos.jar

Dans un deuxième temps, copiez le fichier "StripesResources.properties" du dossier "/lib/" du zip de Stripes que vous avez téléchargé dans "/WEB-INF/classes/".

Configuration du fichier web.xml

- Ouvrez le fichier "web.xml" qui se situe, comme dans toute architecture d'un projet JEE, dans le dossier "/WEB-INF/".
- Rajoutez, dans le fichier "web.xml", le code du fichier XML "stripes.xml" qui se trouve dans le zip de Stripes.

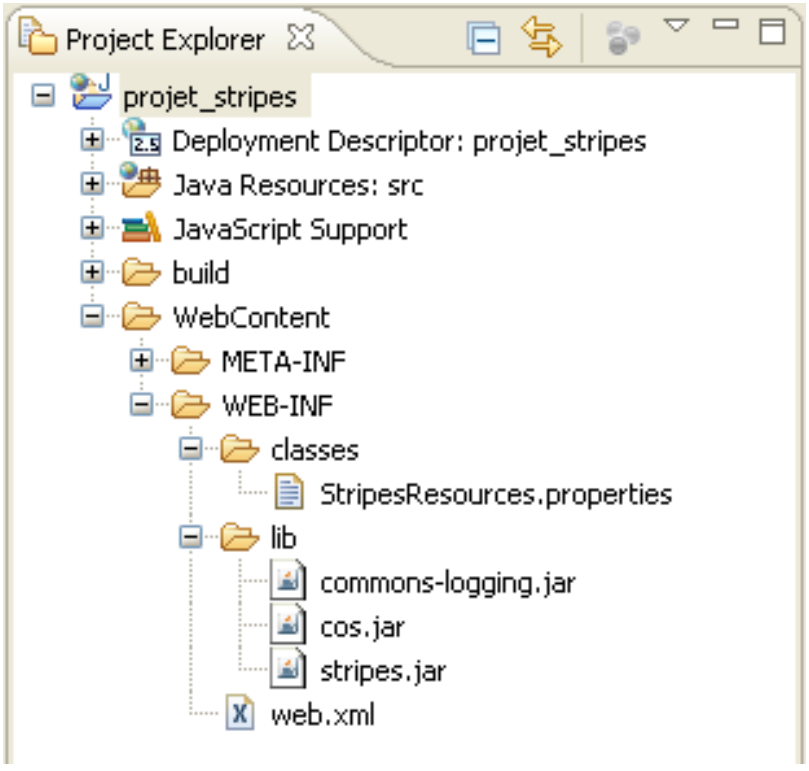
Ce code XML correspond à la configuration de Stripes (détection des ActionBeans, des .jsp, etc...). Pour accélérer la détection des ActionBeans au démarrage de Stripes, ne pas oublier de modifier dans la balise filter/init-param/param-value/ (à l'endroit du "### A MODIFIER ###"), le ou les noms des packages qui contiendront les ActionBeans, séparés par des virgules.

Création d'ActionBean et de JSP

- Vous pouvez dès à présent créer des ActionBean qu'utiliseront vos JSP. Pour cela, il faut respecter les conditions suivantes:
    - Votre classe doit implémenter l'interface ActioBean (**implements ActionBean**)
    - Votre classe doit posséder un attribut de type ActionBeanContext (**private ActionBeanContext context;**)
    - Votre classe doit posséder le getter et le setter pour cet ActioBeanContext.
- Les JSP font appel aux ActionBean afin de réaliser le modèle MVC. Toute JSP doit posséder l'entête suivant :

```
<%@ page language="java" contentType="text/html;charset=UTF-8" %>
<%@ taglib prefix="stripes" uri="http://stripes.sourceforge.net/stripes.tld"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Vous disposez à présent de tous les éléments pour faire du MVC avec Stripes, en utilisant les ActionBean. Le fichier "StripesResources.properties" permet le traitement des erreurs de formulaires (conversion, validation de champs, etc...) et des messages associés à ces erreurs, qui sont entièrement paramétrables.



Configuration

Stripes est conçu de façon à requérir le minimum de configuration possible.

Le web.xml

Une simple configuration du Stripes Filter et du Stripes Dispatcher Servlet est nécessaire. Pour notre projet, nous avons choisis de prendre comme configuration :

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-
app_2_4.xsd">

<!-- Balise pour le titre du projet -->
<display-name>Projet_MSI</display-name>


<!-- ----------------------------------- -->
<!-- ---------- Les Filters ------------ -->
<!-- ----------------------------------- -->
<filter>
    <display-name>Stripes Filter</display-name>
    <filter-name>StripesFilter</filter-name>
    <filter-class>net.sourceforge.stripes.controller.StripesFilter</filter-class>


    <!-- filtre qui map action Bean -->
    <init-param>
      <param-name>ActionResolver.Packages</param-name>
      <param-value>actionBean</param-value>
    </init-param>



   <!-- filtre qui map action Bean context -->
   <init-param>
            <param-name>ActionBeanContext.Class</param-name>
            <param-value>actionBean.TrimaneActionBeanContext</param-value>
   </init-param>


</filter>


<!-- ----------------------------------- -->
<!-- ------- Les Filters Mapping ------- -->
<!-- ----------------------------------- -->
<filter-mapping>
    <filter-name>StripesFilter</filter-name>
    <url-pattern>*.jsp</url-pattern>
    <dispatcher>REQUEST</dispatcher>
</filter-mapping>

<filter-mapping>
    <filter-name>StripesFilter</filter-name>
    <servlet-name>StripesDispatcher</servlet-name>
    <dispatcher>REQUEST</dispatcher>
</filter-mapping>
```

```
<!-- ----------------------------------- -->
<!-- ---------- Les Servelets ---------- -->
<!-- ----------------------------------- -->

<servlet>
        <servlet-name>StickyCaptcha</servlet-name>
        <servlet-class>nl.captcha.servlet.StickyCaptchaServlet</servlet-class>

         <init-param>
             <param-name>width</param-name>
             <param-value>250</param-value>
        </init-param>

        <init-param>
             <param-name>height</param-name>
             <param-value>75</param-value>
        </init-param>

</servlet>

<servlet>
    <servlet-name>StripesDispatcher</servlet-name>
    <servlet-class>net.sourceforge.stripes.controller.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>

<servlet>
    <description></description>
    <display-name>LogActionBean</display-name>
    <servlet-name>LogActionBean</servlet-name>
    <servlet-class>actionBean.LogActionBean</servlet-class>
</servlet>



<!-- ----------------------------------- -->
<!-- ------- Les Servlets Mapping ------ -->
<!-- ----------------------------------- -->
<servlet-mapping>
        <servlet-name>StickyCaptcha</servlet-name>
        <url-pattern>/stickyImg</url-pattern>
</servlet-mapping>

<servlet-mapping>
    <servlet-name>StripesDispatcher</servlet-name>
    <url-pattern>*.action</url-pattern>
</servlet-mapping>

<servlet-mapping>
    <servlet-name>LogActionBean</servlet-name>
    <url-pattern>/LogActionBean</url-pattern>
</servlet-mapping>


<!—La première page du projet à afficher -->
<welcome-file-list>
    <welcome-file>/Processus/Accueil.jsp</welcome-file>
</welcome-file-list>


</web-app>
```

Les Bibliothèques

Puis il faudra que vous placiez stripes.jar dans votre classpath (normalement WEB-INF/lib). Ceci est la seule dépendance à compile-time pour développer avec Stripes. Pour développer et utiliser Stripes vous aurez besoin de copier les fichers libraires ci-dessous fournis avec Stripes dans votre classpath :

- `commons-logging.jar (1.1)` - Apache Commons Logging est utilisé afin de fournir une interface de logging avec une implémentation agnostique.
- `cos.jar` - le package libraire com.oreilly.servlets, grâce à Jason Hunter, est utilisé pour gérer les uploads de fichiers multi-part à partir de formulaires d'envoi

Les fichiers libraires ci-dessus sont fournis avec la distribution de stripes, et ont été testés avec Stripes. Des versions plus récentes peuvent fonctionner, mais vous serez en territoire inconnu.

Les properties

La dernière partie à être plaçée est le fichier StripesResources.properties; il faudra tout simplement le copier dans /WEB-INF/classes. StripesResources.properties est utilisé (par défaut) pour contenir des messages d'erreurs qui sont utilisés pour la validation incluse par défaut dans Stripes et la validation faite dans les ActionBeans, et il doit pouvoir être retrouvé dans le classpath. On peut aussi utiliser les properties pour ne pas écrire en dure les textes dans le code, en utilisant les balises « Label ».

Voici un fragment exemple du fichier :

Example StripesResources.properties

```
# Ecriture des texts.
Titre_page_accueil=Projet de JEE : Stripes
Premiere_Rubrique=Les avantages
Deuxième_Rubrique=Les désavantages


# Validation error messages used by Stripes' built in type converters
converter.number.invalidNumber=The value ({1}) entered in field {0} must be a valid number
converter.byte.outOfRange=The value ({1}) entered in field {0} was out of the range {2} to {3}
converter.short.outOfRange=The value ({1}) entered in field {0} was out of the range {2} to {3}
converter.integer.outOfRange=The value ({1}) entered in field {0} was out of the range {2} to {3}
converter.float.outOfRange=The value ({1}) entered in field {0} was out of the range {2} to {3}
converter.enum.notAnEnumeratedValue=The value "{1}" is not a valid value for field {0}
converter.date.invalidDate=The value ({1}) entered in field {0} must be a valid date
converter.email.invalidEmail=The value ({1}) entered is not
```